



## PGC-104: C++ Programming Advanced

### Course Details

**Course Code:** PGC-104

**Duration:** 5 days

#### Notes:

- This course syllabus should be used to determine whether the course is appropriate for the students, based on their current skills and technical training needs.
- Course content, prices, and availability are subject to change without notice.
- Terms and Conditions apply

*Elements of this syllabus are subject to change.*

#### About this course

This course makes C++ programmers more productive. It focuses on effective use of the language. This course covers advanced language features, describes useful implementation techniques, and reviews object-oriented design guidelines.

#### At Course Completion

After completing this course, students will be able to:

- Identify all the major features of the C++ language, including templates and exceptions
- Discuss issues of inheritance
- Use a powerful set of C++ techniques and idioms for solving problems
- Apply object-oriented design concepts to C++ development

#### Prerequisites

A good working knowledge of the C++ language is absolutely essential before attending this course.

#### Academy IT Pty Ltd

Harmer House  
Level 2, 5 Leigh Street  
ADELAIDE 5000

Email: [sales@academyit.com.au](mailto:sales@academyit.com.au)

Web: [www.academyit.com.au](http://www.academyit.com.au)

Phone: 08 7324 9800

Brian: 0400 112 083

**Experienced Use of C++**

- Using constants and constant pointers
- References, parameters, and variables
- Inline functions and encapsulation
- Templates and initialization

**User-Defined Data Types**

- Defining robust classes
- Constructors and initialization
- Dynamic storage within a class
- Type conversion and array types
- Stand-in classes

**Generic Data Types**

- Collection classes
- Data type vs. data structure
- Class templates
- Names and instantiation
- Specialization

**Class Relationships**

- Inheritance
- Virtual functions, polymorphism, and abstract base classes
- Virtual destructors and base classes
- Constructors

**Memory Management**

- Dynamic object creation
- Vector allocation
- Sharing memory among objects
- Reference counting and cycles
- Swappable objects
- Smart pointers

**Exception Handling**

- Error-handling strategies
- Throwing exceptions
- Inheritance and exceptions

**Mixing C and C++**

- C/C++ source and link incompatibility
- Calling C++ code from C